

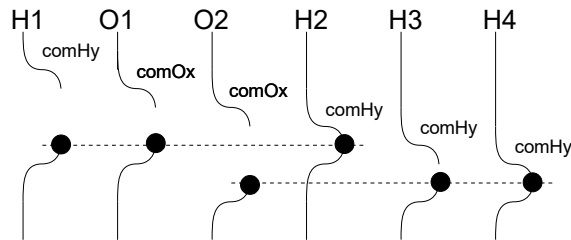
CC4302 Sistemas Operativos – Tarea 5 – Semestre Primavera 2023 - Prof.: Luis Mateu

En esta tarea Ud. deberá implementar una nueva herramienta de sincronización para nThreads basada en la formación de moléculas de agua. Los encabezados de las funciones pedidas son:

```
typedef struct h2o *nH2o;
H2o *nCombineOxy(void *oxy, int timeout);
H2o *nCombineHydro(void *hydro);
void initH2O(void);
void endH2O(void);
```

La función *nCombineOxy* suministra un átomo de oxígeno (*oxy*) y *nCombineHydro* suministra un átomo de hidrógeno (*hydro*). En cuanto haya una llamada pendiente de *nCombineOxy(o)* y 2 llamadas pendientes de *nCombineHydro* con átomos *h1* y *h2*, Ud. debe crear un objeto de tipo *H2o** invocando la función *makeH2o(h1, h2, o)*. La función *nCombineOxy* espera por a lo más *timeout* milisegundos retornando NULL si expiró. Si *timeout* es -1 se espera indefinidamente. Por simplicidad, *nCombineHydro* no tiene *timeout*.

El siguiente diagrama de threads muestra cómo se deben asociar las llamadas de *nCombineOxy* (abreviado *comOx*) con las llamadas de *nCombineHydro* (*comHy*).



Restricciones

Ud. debe programar las funciones solicitadas como herramientas de sincronización nativas de nThreads, es decir usando operaciones como *START_CRITICAL*, *setReady*, *suspend*, *schedule*, *nth_programTimer*, *nth_cancelThread*, etc. Ud. no puede implementar la API solicitada en términos de otras herramientas de sincronización pre-existentes en nThreads (como semáforos, mutex, condiciones o mensajes). Ejemplos de la solución que se espera de Ud. son la implementación de los semáforos, mutex, condiciones y mensajes de nThreads (en los archivos *nKernel/sem.c*, *nKernel/mutex-cond.c* y *nKernel/nmsgs.c*).

Instrucciones

Descargue *t5.zip* de U-cursos y descomprímalo. Copie su archivo *h2o.c* con la solución de su tarea 4 a este mismo directorio. Modifique *h2o.c*

implementando los timeouts que no se pedían en la tarea 4. Ejecute el comando *make* sin parámetros en el directorio *T5* para recibir instrucciones sobre cómo compilar y probar su solución, los requisitos que debe cumplir para aprobar la tarea y cómo entregar su tarea por U-cursos. Además se explica cómo puede probar sus tareas 1, 2 y 3 usando nThreads como implementación de pthreads.

Ayuda

Use 2 colas de tipo NthQueue: una para los threads que suministran oxígeno y la otra para los threads que suministran hidrógeno. Cuando un thread necesite esperar, coloque el átomo suministrado en el campo *ptr* del descriptor de thread. Use ese mismo campo para almacenar la dirección de la molécula de H2O cuando se forme. Al retornar retorne el valor en *ptr*.

Necesita distinguir entre nthreads que invocaron *nCombineOxy* con *timeout* (*timeout*>0) y las que lo hicieron sin *timeout* (*timeout*<0). Para las primeras use el estado *WAIT_H2O_TIMEOUT*. Para las segundas use el estado *WAIT_H2O*. Es importante porque cuando se invoca *nCombineOxy*, el nthread en espera en un *nCombineOxy* con *timeout* está configurado para despertarse con *nth_programTimer*. Si su estado continúa en *WAIT_H2O_TIMEOUT* su *timeout* debe cancelarse con *nth_cancelThread* (vea la [clase auxiliar de mensajes con timeout](#)).

Observe que *nth_programTimer* recibe el *timeout* en nanosegundos, mientras que *nCombineOxy* recibe el *timeout* en milisegundos. Para convertir *timeout* en milisegundos a nanosegundos, use la expresión *timeout*1000000LL* para que el resultado sea de tipo long long, de otro modo el tipo sería int y habría desborde.

Recuerde que el segundo parámetro de *nth_programTimer* es el puntero a una función que se invocará cuando expire el *timeout*. Necesitará suministrar una función que borre el thread de la cola de oxígeno con *nth_delQueue* y dejar *ptr* en NULL. Tenga cuidado porque esa función se invoca dentro de una rutina de atención de señales. Haga cosas simples en esa función.

Entrega

Ud. solo debe entregar por medio de U-cursos el archivo *h2o.zip* generado por *make zip*. **Recuerde descargar el archivo que subió, descargar nuevamente los archivos adjuntos y volver a probar la tarea tal cual como la subió a U-cursos.** Solo así estará seguro de no haber entregado archivos incorrectos. Se descuenta medio punto por día de atraso. No se consideran los días de receso, sábado, domingo o festivos.